

Introduction to Wavelets in Scilab

Anuradha Amrutkar

Indian Institute of Technology Bombay
anuradha.amrutkar@gmail.com

National Workshop on Scilab
Fr. C. Rodrigues Institute of Technology, Vashi
December 3, 2010



Introduction



- The word **WAVELET** literally means small wave.
- Wavelets are localised waves and they extend not from $-\infty$ to $+\infty$ but only for a finite duration of time.



Introduction



- Since waves extend over the entire space, they do not need any shift parameter.
- Thus, a Fourier Transform maps 1-D time signals to 1-D frequency signals, whereas
- The wavelet transform maps 1-D time signals to 2-D scale(frequency) and shift parameter signals.



Example1



Example1

- Let us see a program which finds out the approximate coefficients and detailed coefficients of a given signal.



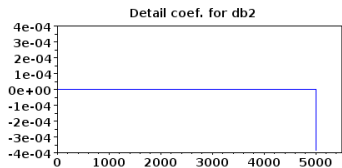
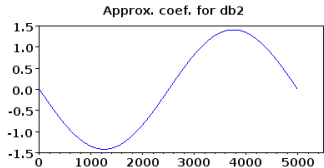
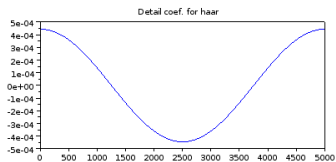
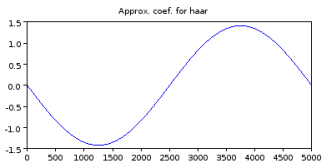
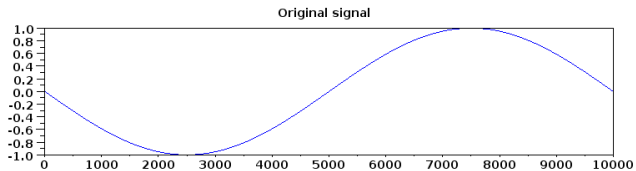


In this Example:

1. `x=linspace(-%pi,%pi,10000);`
2. `s=sin(x);` //Constructs and Elimentary **sine wave** signal
3. `[ca1,cd1] = dwt(s,'haar');` // Perform single-level discrete wavelet transform of **"s"** by **"haar"**.
4. The Graph of Apporoximate co-efficients(cA) and Detailed co-efficient(cD) is Plotted using the **plot()** command
5. The above procedure is repeated for **"db2"** type of wavelet.



Example1:dwt.png

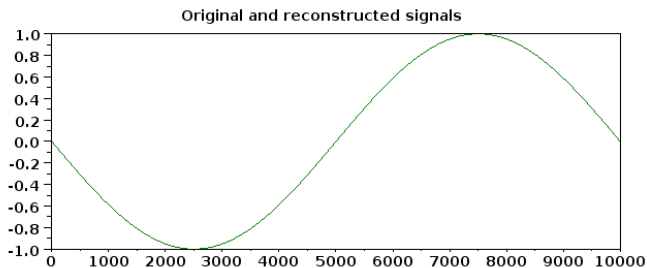
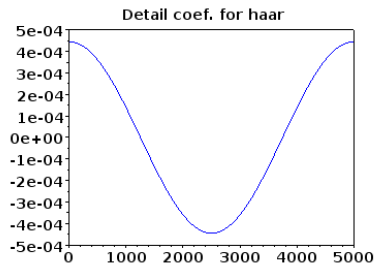
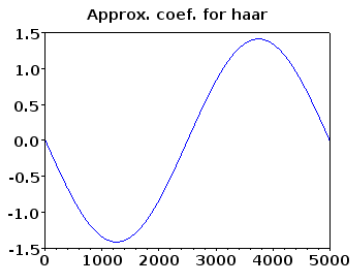




In this Example:

1. Steps 1,2 and 3 are same as above.
2. `ss = idwt(ca1,cd1,'haar');` //Perform single-level inverse discrete wavelet transform, illustrating that idwt is the inverse function of dwt.
3. The Graph of Approximate co-efficients(cA) and Detailed co-efficient(cD) is Plotted using the `plot()` command





Commands : dwt & idwt



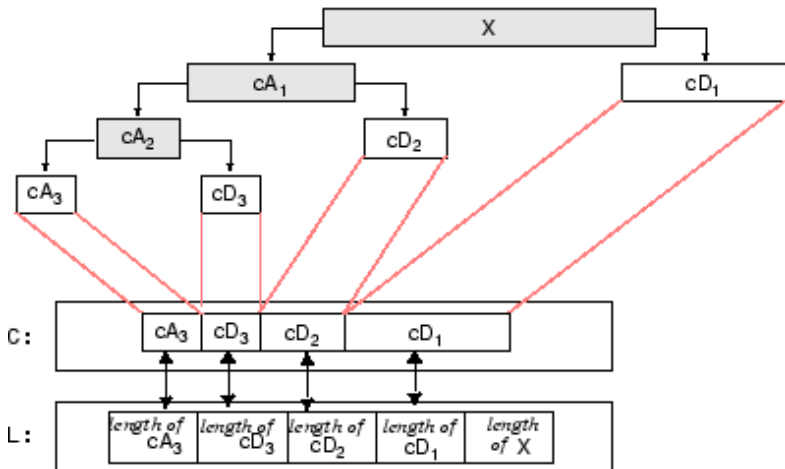
- `dwt`: Discrete Fast Wavelet Transform
 - `dwt` is for discrete fast wavelet transform with the signal extension method optional argument.
 - As output it gives values of `cA` : Approximate co-efficients and `cD` : Detailed co-efficients
 - For Syntax Detailed help see type "`help dwt`"
- `idwt`: Inverse Discrete Fast Wavelet Transform
 - `idwt` is for inverse discrete fast wavelet transform.
 - Coefficient could be void vector as `[]` for `cA` or `cD`.
 - As output it gives a Reconstructed Vector
 - For Syntax Detailed help see type "`help idwt`"





Let us Revise the Decomposition Diagram for the wavelets:

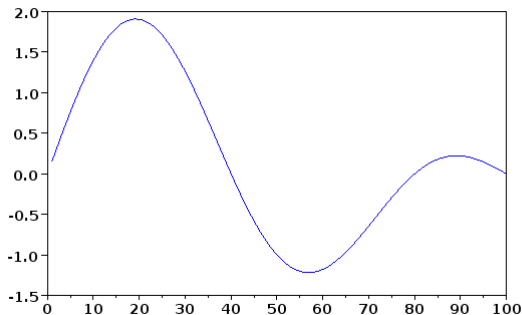
Decomposition:





In this Example:

1. `s=[1:100];`
2. `l_s = length(s)`
3. `a = sin(2*%pi*s/100)+sin(3*%pi*s/100); //Constructs and Elimentary sine wave signal`





The coefficients of all the components of a third-level decomposition (that is, the third-level approximation and the first three levels of detail) are returned concatenated into one vector, C.

Vector L gives the lengths of each component.

```
4. [C,L] = wavedec(a,3,'haar');
```





To extract the level 3 approximation coefficients from C, type:

```
5. cA3 = appcoef(C,L,'haar',3);
```

To extract the levels 3, 2, and 1 detail coefficients from C, type

```
6. cD3 = detcoef(C,L,3);
```

```
7. cD2 = detcoef(C,L,2);
```

```
8. cD1 = detcoef(C,L,1);
```

The above can be written in one command as:

```
9. [cD1,cD2,cD3] = detcoef(C,L,[1,2,3]);
```





To reconstruct the level 3 approximation from C, type

```
10. A3 = wrcoef('a',C,L,'haar',3);
```

To reconstruct the details at levels 1, 2, and 3, from C, type

```
11. D1 = wrcoef('d',C,L,'haar',1);
```

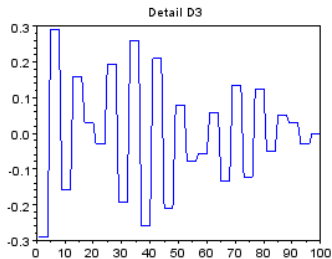
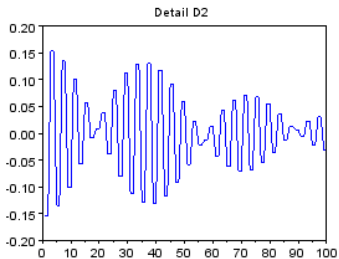
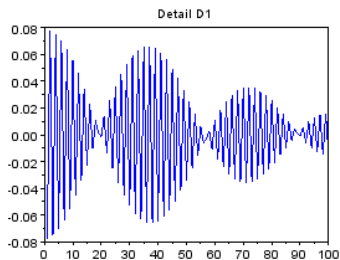
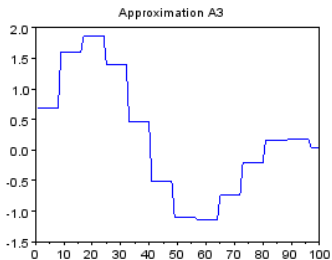
```
12. D2 = wrcoef('d',C,L,'haar',2);
```

```
13. D3 = wrcoef('d',C,L,'haar',3);
```





Display the results of a multilevel decomposition





To reconstruct the original signal from the wavelet decomposition structure, type

```
14. A3 = waverec(C,L,'haar');
```

Of course, in discarding all the high-frequency information, we've also lost many of the original signal's sharpest features.

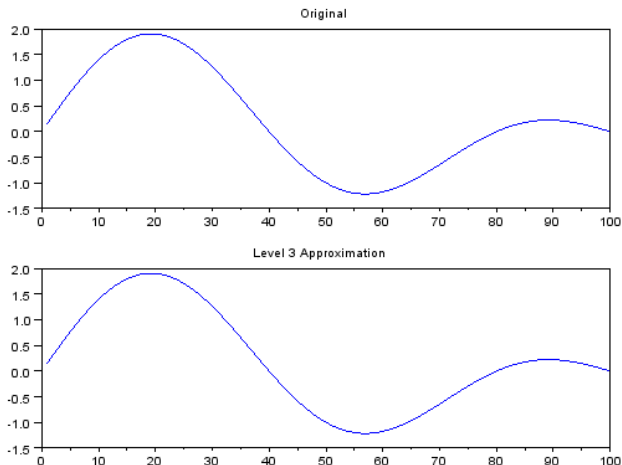


Original Vs Approximate



Original Vs Approximate

To compare the approximation to the original signal, type



Commands Used



The commands used in the Multi-level Decomposition and Construction of Approximate and Detailed Coefficients are:

- ★ `wavedec`: Multiple Level Discrete Fast Wavelet Transform
- ★ `waverec`: Multiple Level Inverse Discrete Fast Wavelet Transform
- ★ `appcoef`: One Dimension Approximation Coefficient Reconstruction
- ★ `detcoef`: One Dimension Detail Coefficient Extraction
- ★ `wrcoef`: Restruction from single branch from multiple level





Please type **help command_name** to see the Usage, Description and Examples for that particular command.



Further Exploration



Optimal de-noising requires a more subtle approach called thresholding.



Optimal de-noising requires a more subtle approach called thresholding.

This involves discarding only the portion of the details that exceeds a certain limit.



Thank You!!



- Thank You!

